





A5

GENERAL PROCESSOR FOR IMAGE PROCESSING

Patent number: JP6223166
Publication date: 1994-08-12
Inventor: KUROKAWA NOBORU; others: 01
Applicant: SONY CORP
Classification:
 - **International:** G06F15/66; G06F15/16; G06F15/68
 - **European:**
Application number: JP19930011807 19930127
Priority number(s):

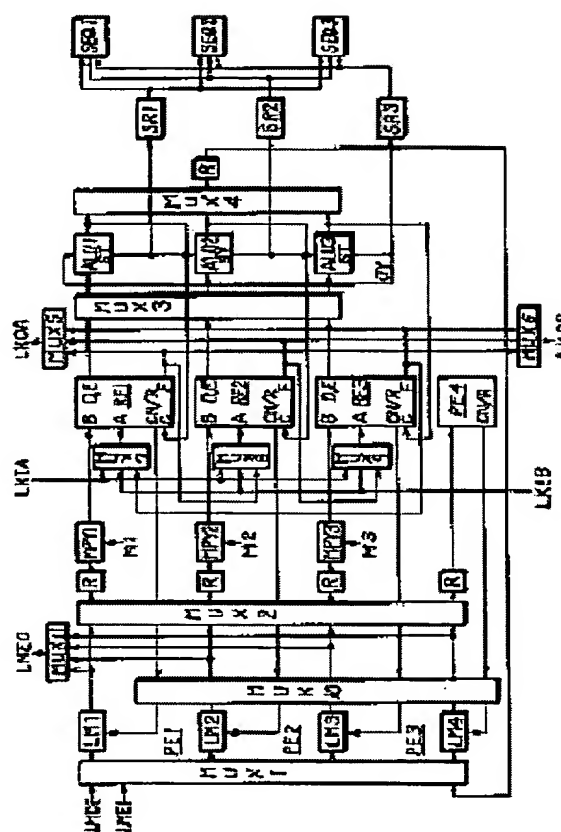
Also published as:

 EP0610688 (A2)
 US5602727 (A1)
 EP0610688 (A3)
 EP0610688 (B1)

Abstract of JP6223166

PURPOSE: To easily expand product sum arithmetic processing and to facilitate processing as to plural image memories by incorporating data in arithmetic processing steps of mutually different processor elements in its own processor element.

CONSTITUTION: An input switching means MUX2 determines which of multipliers MPY1-MPY3 respective outputs of a line memory LM are supplied to. The outputs of address counters CN provided for register files RF1-RF3 are utilized as line memory specification addresses at this time. The multiplication results are loaded from a terminal B and image data selected by input switching means MUX7, MUX8, and MUX9 are loaded from a terminal A. An input switching means MUX4 selects which of line memories LM one of arithmetic processing results is written in through a register R for a buffer and in this case, the input switching means MUX3 and MUX1 operate at the same time.



Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-223166

(43)公開日 平成 6 年(1994) 8 月12日

(51)IntCl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/66		J 8420-5L		
15/16	3 9 0	Z 9190-5L		
15/68	4 0 0	J 9191-5L		

審査請求 未請求 請求項の数 7 O L (全 16 頁)

(21)出願番号 特願平5-11807

(22)出願日 平成 5 年(1993) 1 月27日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川 6 丁目 7 番35号

(72)発明者 黒川 登

東京都品川区北品川 6 丁目 7 番35号 ソニ
ー株式会社内

(72)発明者 安藤 辰伸

東京都品川区北品川 6 丁目 7 番35号 ソニ
ー株式会社内

(74)代理人 弁理士 山口 邦夫 (外 1 名)

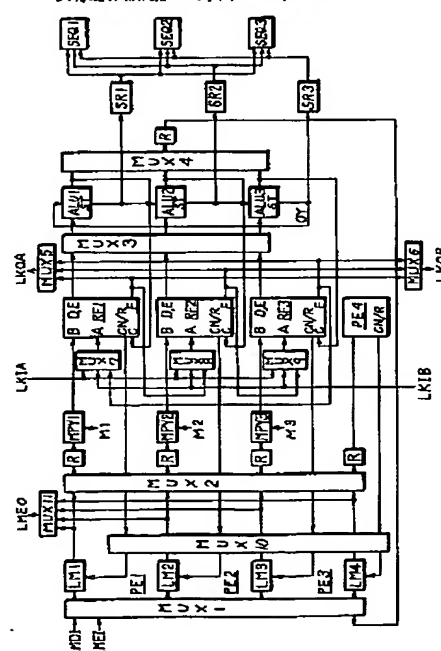
(54)【発明の名称】 画像処理用汎用プロセッサ

(57)【要約】

【目的】画像処理に対して汎用化できるプロセッサを実現する。

【構成】積和演算機能を有する複数のプロセッサエレメント P E で構成され、これら複数のプロセッサエレメント P E 同士は互いに別のプロセッサエレメント P E の任意の演算処理ステップにおけるデータを、自己のプロセッサエレメント P E における任意の処理ステップに取り込むことができるように、入力切り替え手段 M U X が設けられる。プロセッサエレメント P E はデータライト用ラインメモリ L M、その出力データを乗算する乗算器 M P Y、乗算出力などを取り込むレジスタファイル R F、レジスタファイル R F の出力を演算する A L U で構成される。これで、パイプライン処理ができ、また個々の処理素子の前段に入力切り替え手段 M U X が設けられているのでプロセッサエレメント間でのデータ授受ができる。

画像処理用汎用プロセッサ P U の具体例



1

【特許請求の範囲】

【請求項1】 積和演算機能を有する複数のプロセッサエレメントで構成され、

これら複数のプロセッサエレメント同士は互いに別のプロセッサエレメントの任意の演算処理ステップにおけるデータを、自己のプロセッサエレメントにおける任意の処理ステップに取り込むことができるようになされたことを特徴とする画像処理用汎用プロセッサ。

【請求項2】 上記プロセッサエレメントは、データをライトするラインメモリ、ラインメモリの出力データを乗算する乗算器、乗算出力などを取り込むレジスタファイル、レジスタファイルの出力を演算する累算器で構成され、
個々の処理素子の前段にはデータセレクト用の入力切り替え手段が設けられたことを特徴とする請求項1記載の画像処理用汎用プロセッサ。

【請求項3】 上記データはバイブライン処理によって積和演算処理がなされるようになされたことを特徴とする請求項2記載の画像処理用汎用プロセッサ。

【請求項4】 (5×5)サイズの積和演算処理を行うとき、3個の画像処理用汎用プロセッサが使用され、そのうちの2個の画像処理用汎用プロセッサがライン内のデータ演算処理に使用され、
残りの画像処理用汎用プロセッサがライン間のデータ演算処理に使用されるようになされたことを特徴とする画像処理用汎用プロセッサ。

【請求項5】 2以上の画像メモリのデータ同士を演算処理するに当たり、

少なくともそれぞれの画像メモリのデータを入力する2以上のプロセッサエレメントが使用され、
それぞれのプロセッサエレメントのレジスタファイルにロードされたデータは、それぞれの累算器の前段に設けられた入力切り替え手段を制御することによって他方のプロセッサエレメントに設けられた累算器に取り込むことができるようにしたことを特徴とする画像処理用汎用プロセッサ。

【請求項6】 2つの上記プロセッサエレメントにそれぞれ設けられた累算器におけるその演算結果であるステータス情報がそれぞれ結合されて、次に処理すべき条件分岐用マイクロプログラムのアドレス情報として使用されるようになされたことを特徴とする請求項5記載の画像処理用汎用プロセッサ。

【請求項7】 画像処理用汎用プロセッサを画像メモリの制御回路として使用するに当たり、
画像処理用汎用プロセッサに設けられた2つのプロセッサエレメントが上記画像メモリに対するアドレス発生器として使用され、

上記画像メモリよりリードされた画像データは一旦上記画像処理用汎用プロセッサに設けられたラインメモリにストアされ、

2

このデータストアに際してそのコントロールが画像処理用汎用プロセッサに設けられた残りのプロセッサエレメントに基づいて行われるようになされたことを特徴とする画像処理用汎用プロセッサ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、画像データのフィルタリング処理などに適用して好適な画像処理用汎用プロセッサに関する。

【0002】

【従来の技術】画像処理用のプロセッサとしてはその処理目的にのみ対応できるように専用化されている。例えば2次元画像を処理する並列処理プロセッサとしては、「特開昭62-208158号公報」などが知られている。

【0003】このLSIは画像処理用のアルゴリズムを機能的に回路化して1個のLSIに集積したものであり、特定の積和演算処理などの特定のアルゴリズムに対してはパラメータ入力により高速に処理できる。

【0004】また、マイクロプログラム制御によるプロセッサとしては、「特開昭63-118885号公報」などが知られている。このLSIは局所処理用に4つのプロセッサを持たしたもので、各プロセッサがマイクロプログラムに基づいてマルチプロセッサとして動作するように構成されている。

【0005】

【発明が解決しようとする課題】このように従来の画像処理用のプロセッサは何れも専用のプロセッサであるから、特定のアルゴリズムを処理する場合には高速化が可能であるのに対して、特定のアルゴリズム以外のアルゴリズムはこれを処理できない。複数のプロセッサを組み合わせ使用したとしてもプロセッサ間のデータ通信ができないため、これもまた特定のアルゴリズムのみに対応できるマルチプロセッサ構成となっている。

【0006】そのため、汎用性が全くないから、例えばこのプロセッサを(3×3)サイズの積和演算処理から、(5×5)サイズ以上の積和演算処理に拡張することもできなければ、複数のプロセッサを用いて分割処理することによって処理の高速化を達成することもできない。

【0007】また、プロセッサ間でのデータ通信ができれば、例えば2つの画像メモリのデータ同士を演算処理することができるので、2つの画像データを論理和したりする演算を簡単に行うことができる。さらには、レジスタファイルのアドレス機能を利用すれば画像メモリに対するアドレス制御用の回路などにも適用することが可能になる。

【0008】そこで、この発明はこのような従来の課題を解決したものであって、汎用化を実現した画像処理用汎用プロセッサを提案するものである。

【0009】

【課題を解決するための手段】上述の課題を解決するため、第1の発明においては、積和演算機能を有する複数のプロセッサエレメントで構成され、これら複数のプロセッサエレメント同士は互いに別のプロセッサエレメントの任意の演算処理ステップにおけるデータを、自己のプロセッサエレメントにおける任意の処理ステップに取り込むことができるようになされたことを特徴とするものである。

【0010】第2の発明においては、(5×5)サイズの積和演算処理を行うとき、3個の画像処理用汎用プロセッサが使用され、そのうちの2個の画像処理用汎用プロセッサがライン内のデータ演算処理に使用され、残りの画像処理用汎用プロセッサがライン間のデータ演算処理に使用されるようになされたことを特徴とするものである。

【0011】第3の発明においては、2以上の画像メモリのデータ同士を演算処理するに当たり、少なくともそれぞれの画像メモリのデータを入力する2以上のプロセッサエレメントが使用され、それぞれのプロセッサエレメントのレジスタファイルにロードされたデータは、それぞれの累算器の前段に設けられた入力切り替え手段を制御することによって他方のプロセッサエレメントに設けられた累算器に取り込むことができるようにしたことを特徴とするものである。

【0012】第4の発明においては、画像処理用汎用プロセッサを画像メモリの制御回路として使用するに当たり、画像処理用汎用プロセッサに設けられた2つのプロセッサエレメントが上記画像メモリに対するアドレス発生器として使用され、上記画像メモリよりリードされた画像データは一旦上記画像処理用汎用プロセッサに設けられたラインメモリにストアされ、このデータストアに際してそのコントロールが画像処理用汎用プロセッサに設けられた残りのプロセッサエレメントに基づいて行われるようになされたことを特徴とするものである。

【0013】

【作用】第1の発明について説明すると、図1に示すように、プロセッサエレメントPE1、PE2、PE3は、それぞれ図2のようにデータをライトするラインメモリLM、ラインメモリLMの出力データを乗算する乗算器MPY、乗算出力などを取り込むレジスタファイルRF、レジスタファイルRFの出力を演算する累算器ALUで構成される。

【0014】個々の処理素子の前段にはデータセレクト用の入力切り替え手段MUXが設けられ、この入力切り替え手段MUXを適宜制御することによってプロセッサエレメント同士の任意の処理ステップの段階でデータの取り込みができるようになる。例えば入力切り替え手段MUX1を制御することによってデータはどのラインメモリLM1～LM4にも取り込むことができるし、入力

切り替え手段MUX2とMUX10を制御することによって、ラインメモリLM1～LM4のデータをどの乗算器MPX1～MPX3にも取り込むことができる。

【0015】その結果、他の発明として開示したようにプロセッサエレメント間のデータ取り込みは勿論のこと、複数のプロセッサエレメントで構成されるプロセッサ（以下プロセッサユニットPUという）同士のデータ通信も可能になる。

【0016】

【実施例】続いて、この発明に係る画像処理用汎用プロセッサの一例を上述した画像処理に適用した場合につき、図面を参照して詳細に説明する。

【0017】図1はこの発明の基本構成を示し、図のようにそれ自身計算機能を有したプロセッサエレメントPEが複数、この例では3個PE1～PE3使用されて汎用プロセッサPUが構成される。

【0018】プロセッサエレメントPE同士はその任意の処理ステップで互いの画像データを入力することができるようになされているが、図1には説明の都合上その接続関係が省略され、エレメント同士は互いに独立したものであるとして図示されている。プロセッサエレメントPE1～PE3は同一構成であるので、プロセッサエレメントPE1についてその構成を説明する。

【0019】図1に示すプロセッサエレメントPE1において、LMはラインメモリである。本例では画像データを処理することを前提としているので、メモリの単位はラインである。ラインメモリLMよりリードされた画像データは乗算器MPYにて所定の係数Mと乗算される。乗算結果はレジスタファイルRFの所定のアドレスにストアされる。

【0020】レジスタファイルRFよりリードされた複数の画像データは累算器ALUにて積算処理され、その結果は再びレジスタファイルRFにストアされる。レジスタファイルRFにストアされた積算結果はリードされて別のプロセッサエレメントPE2若しくはPE3のレジスタファイルにロードされる。

【0021】累算器ALUにおける乗算した結果であるステータス情報（キャリー／オーバーフロー／サイン／ゼロなど）はステータスレジスタSRに一旦ストアされると共に、そのステータス情報（1ビット）はシーケンサSEQに格納されたマイクロプログラムのアドレスの一部として使用される。

【0022】シーケンサSEQ1～SEQ3は対応するプロセッサエレメントPE1～PE3を制御するために設けられたもので、これに内蔵されたマイクロプログラムによって演算処理モードが決定される。

【0023】ラインメモリLMには入力画像データの他、他の汎用プロセッサPUより出力された演算処理結果が格納される場合があり、この演算処理結果は最終的には別の画像メモリにストアされる。したがって、ライ

5

ンメモリLMの出力は上述した乗算器MPYへの入力データとして使用されるほか最終データとして外部にも出力される。

【0024】上述したプロセッサエレメントPE1~PE3はその処理途中の段階(ステップ)で、任意のプロセッサエレメントPE1~PE3からの入力画像データ若しくは処理データを受け取ることができるになっている。そのための具体的な構成の一例を図2に示す。

【0025】図2において、ラインメモリLMは4個使用され、乗算器MPYは3個使用される。乗算器MPYが3個使用される結果、後段に続くレジスタファイルRF、累算器ALUも、ステータスレジスタSRも、そしてシーケンサSEQもそれぞれ3個使用される。

【0026】このような個数のときを例示すると、まずMUX1は入力切り替え手段であって、これにはこの例では3つの入力データが供給され、そのうちの任意の入力データを選択して出力させることができるようになされている。入力データの1つであるLMDIは外部に設けられた画像メモリ(図示はしない)からの入力画像データであり、LMEIは別の汎用プロセッサPUより出力された画像データを取り扱うときの入力画像データである。そして、もう1つの入力データは汎用プロセッサPU自身の演算処理結果の画像データである。

【0027】ラインメモリLMのそれぞれの出力は入力切り替え手段MUX2によってどの乗算器MPY1~MPY3に供給されるかが決まる。このときのラインメモリ指定アドレスはレジスタファイルRF1~RF3に設けられたアドレスカウンタCNの出力が利用される。このアドレスカウンタCNのアドレスによってどのラインメモリLM1~LM4でもその出力を指定して対応する乗算器MPY1~MPY3に入力させることができる。Rはバッファ用のレジスタである。

【0028】レジスタファイルRF1~RF3は3つの入力端子A, B, Cと3つの出力端子D, E, FとアドレスCNの7端子構成であって、これらの入出力処理およびアドレス指定は同時に実行できるようになっている。

【0029】乗算結果はB端子からロードされ、A端子からは入力切り替え手段MUX7, MUX8, MUX9で選択された画像データがロードされる。入力切り替え手段MUX3はレジスタファイルRF1~RF3の画像データ(D, E端子)をどの累算器ALU1~ALU3に与えるかを選択するためのものである。

【0030】入力切り替え手段MUX4は演算処理結果の何れかをバッファ用のレジスタRを介してラインメモリLMの何れかにライトするかを選択するためのもので、この場合には入力切り替え手段MUX3とMUX1がそれぞれ同時に動作する。ALU1~ALU3の出力はさらにレジスタファイルRF1~RF3のC端子を通じてロードされる。

6

【0031】入力切り替え手段MUX5とMUX6とはレジスタファイルRF1~RF3の出力画像データ(F端子)を選択して別の汎用プロセッサに渡すためのもので、入力切り替え手段MUX5およびMUX6で選択された出力画像データLKOA, LKOBは別の汎用プロセッサにおける入力画像データ(処理中の画像データ)LKIA, LKIBとなる。

【0032】入力切り替え手段MUX10はラインメモリLMの画像データを別の画像メモリなどにライトするときなどのときに外部出力制御用として使用されるものであって、プロセッサエレメントPE4によって制御される。すなわち、プロセッサエレメントPE4に内蔵されたマイクロプログラムによって生成されたアドレスカウンタ出力CNに基づいて入力切り替え手段MUX10が動作して特定のラインメモリLMが選択される。プロセッサエレメントPE4は図示はしないが、他の入力切り替え手段MUX1~MUX11も制御する。

【0033】ラインメモリLMよりリードされた画像データは入力切り替え手段MUX11によって何れかの画像データのみが選択されて出力画像データLMEOとなされる。

【0034】このようにデータ処理中はレジスタファイルRF1~RF3からのアドレスカウンタ出力CNによって入力切り替え手段MUX10が制御されるのに対して、最終出力画像データを外部に出力するときにはプロセッサエレメントPE4からのアドレスカウンタ出力CNによって入力切り替え手段MUX10が制御されるようになっている。

【0035】プロセッサエレメントPE4は上述したプロセッサエレメントPE1~PE3の構成要素のうちレジスタファイルRF, 累算器ALU, ステータスレジスタSRおよびシーケンサSEQの構成要素によって成り立っている。

【0036】レジスタファイルRF1~RF3はその縦方向に並んだ隣りのレジスタファイルにのみ出力画像データを受け渡すことができるようになされており、出力画像データは上述した入力切り替え手段MUX7, MUX8, MUX9を選択することによって行われる。したがって、入力切り替え手段MUX7, MUX8, MUX9では外部からの画像データLKIA, LKIBと前段のF端子からの出力画像データの計3つの画像データが選択される。

【0037】累算器ALUの演算処理結果であるステータス情報はそれぞれ縦方向に配列された次段の累算器ALUに伝達されると共に、ステータスレジスタSR1~SR3にも供給されて一時的にストアされる。このステータス情報(1ビット)は次段に設けられたシーケンサSEQ1~SEQ3の全てに供給され、3つのステータス情報を結合することによって下位3ビットのアドレスとして使用される。これでシーケンサSEQ1~SEQ

7

3のそれぞれに格納されたマイクロプログラムに対するアドレス(3ビット分)が指定される。このような結合処理を行うことによってパイプライン処理の高速化を達成している。

【0038】図2に示したような構成の汎用プロセッサPUを使用して画像処理用プロセッサが構成される。図3はその基本構成を示す。

【0039】図3は6個の汎用プロセッサ(以下プロセッサユニットという)PUを使用して画像処理用プロセッサを構成した場合で、実線は入力画像データの経路、¹⁰ 破線は出力画像データの経路をそれぞれ示す。鎖線は処理用データの経路(流れ)であって、横方向に並んだプロセッサユニット(PU1, PU2), (PU3, PU4), (PU5, PU6)同士の間では図のようにデータが流れる(鎖線図示)のに対し、縦方向に並んだプロセッサユニット(PU1, PU3, PU5), (PU2, PU4, PU6)同士では鎖線矢印方向のみデータが伝達されるように組み合わせられている。

【0040】したがって、1つの画像メモリに蓄積された1枚の画像データに対して例えばエッジ検出用のフィルタリング処理(3画素×3ライン、5画素×5ライン²⁰などの積和演算処理)を施す場合には図4のように構成されることになる。

【0041】図4は積和演算処理用に構成した第2の発明に係る画像処理用汎用プロセッサの例で、特に同図は5×5の積和演算処理を実現するときの具体例である。プロセッサユニットには4個のラインメモリしか搭載していないので、3個のプロセッサユニットが使用され、そのうち積和演算用に2個使用されると共に、演算結果が別のプロセッサユニットにロードされる。³⁰

【0042】図は画像メモリIMを上下に2分割し、それぞれを同時に処理して処理結果を別の画像メモリOMに書き込むようにした場合の例であって、その場合には図のように3個の汎用プロセッサを2組(PU1-PU2-PU4), (PU6-PU5-PU3)に分けてそれぞれをプロセッサユニットPUA, PUBとして並列的な画像処理を行う。

【0043】このとき、画像処理の流れは矢印の通りのパイプライン処理であって、画像データIMAはプロセッサユニットPU1とPU2とに入力され、他方の画像⁴⁰ データIMBはプロセッサユニットPU6とPU5に入力される。そして、プロセッサユニットPU4とPU3から画像処理された出力画像データOMA, OMBが出力される。

【0044】画像データDijの一例を図5に、使用されるフィルタ係数Mijを図6に示す。(5×5)の画像処理は第1に画像データとフィルタ係数の乗算処理が各ラインa~eごとに行われ、第2に各ラインa~eごとに得られた乗算出力に対する加算処理が行われる。第1の乗算処理はプロセッサユニットPU1とPU2とで行わ⁵⁰

8

れ、第2の加算処理はプロセッサユニットPU6で行われる。

【0045】図7~図9は乗算処理を説明するための図である。図7に示すプロセッサユニットPU2では3つのラインメモリLM1~LM3が使用されてラインa~cの画像データが順次ロードされる(時点i0~i2)。

【0046】プロセッサユニットPU1では2個のラインメモリLM4とLM5が使用されて対応するラインdとeの画像データが順次ロードされる(時点i3~i4)。5個のラインメモリLMの全てに画像データがロードされると、図8に示すようにこれらの画像データがラインごとに順次1画素ずつ読み出され、対応する乗算器MPY(MPY1~MPY3)において乗算処理が行われる。

【0047】この乗算処理の一例をラインaについて説明すれば図10のようになる。この図10における処理ステップaがプロセッサユニットPU2における乗算器MPY1の乗算処理例である。ここで、図10の時点t0が図8に示す時点i5に相当する。

【0048】このような乗算処理と同時に、図8のようにプロセッサユニットPU1の空いているラインメモリLM3に次のラインfの画像データがロードされると共に、プロセッサユニットPU2に対しても空いているラインメモリLM4に、プロセッサユニットPU1の初段に位置するラインメモリLM1の画像データがロードされる。時点i6は1ライン処理後の時点t0に対応する。

【0049】このように新しいラインに対する画像データのロードと、空いているラインメモリへの画像データのシフト処理を行うのは、順次1ラインずつ画像データを更新して(5×5)の積和演算処理を実行する必要があるからである。

【0050】プロセッサユニット同士の画像データの授受は図2でも説明したようにLME0とLMEIを利用して行われる。

【0051】ラインa~eを用いた積和演算処理が終了すると、今度はラインを1ラインシフトして図9のようにラインb~fを用いた積和演算処理が行われる。そして、次の処理に備えるべくラインgのロードとラインeのシフト処理が行われる。このような処理が最後のラインまで行われて(5×5)の積和演算処理が終了する。

【0052】図10に示すように、処理ステップaでは乗算器MPYでの画素ごとの乗算処理が行われ、順次乗算出力が順次レジスタファイルRFにロードされると共に、5画素分の乗算処理が終了すると、処理ステップbに示すようにそれらの乗算出力が順次リードされて累算器ALU1で加算処理が行われる。つまり、累算器ALU1では5画素分の乗算出力に対する加算処理が行われ、その最終結果の加算出力R10(ラインaの処理であるためR10aとして示す)が同じプロセッサエレメントPE1内のレジスタファイルRF1のC端子を通じ

てロードされる。

【0053】加算出力R10aは次のタイミング(図10の処理ステップc)にはプロセッサユニットPU4のプロセッサエレメントPE1に設けられたレジスタファイルRF1にロードされて、次のライン間における加算処理に備えられる。このときは図2に示すようにLKOA, LKIA(若しくはLKOB, LKIB)を介して処理データの入出力が行われる。

【0054】処理データの流れは図4に示すように一方(プロセッサユニットPU2→PU1→PU4)であるから、処理データは図11のような流れに沿って最終のプロセッサユニットPU4に伝達される。

【0055】図12を参照して説明する。時点t9になると各ラインa~eの加算出力R10a~R10eが揃うので、時点t10でR10aをプロセッサユニットPU4に移し、これと同時にプロセッサユニットPU2のレジスタファイルRF1にはR10eを、同じレジスタファイルRF2にはR10dをロードする。

【0056】次の時点t11では加算出力R10bをプロセッサユニットPU4に移し、これと同時にプロセッサユニットPU2のレジスタファイルRF2には加算出力R10dを移すというような順序で、全ての加算出力R10a~R10eが順次プロセッサユニットPU4に移される。

【0057】プロセッサユニットPU4では以下のようなライン間の加算処理が実行される。図12と図13を参照して説明する。

【0058】時点t14になると5ライン分の加算出力R10a~R10eが揃うので、これら加算出力R10a~R10eが順次累算器ALU1に供給されて順次加算処理が行われる(図12時点t15~t18)。最終加算結果R00は画素D00におけるフィルタリング処理された画像データとなり、これは図13に示す入力切り替え手段MUX4→レジスタR→MUX1を経てラインメモリLM1にライトされる。

【0059】1ライン分の最終加算結果の画像データRijがラインメモリLM1にストアされると、次のラインの最終加算結果である画像データRijが今度は次のラインメモリLM2にライトされる。これと同時にラインメモリLM1がリードされて図4に示す外部画像メモリOMに画像処理後のデータとしてストアされる。

【0060】ここで、どのラインメモリLMに画像データRijをライトするかはレジスタファイルRF1のアドレスカウンタCNの出力に基づいて制御されるのに対して、ラインメモリLMにストアされた画像データRijを外部メモリOMにストアするときにはプロセッサエレメントPE1の出力に基づいて制御される。

【0061】図14はフィルタリング係数(マスクデータ)としてケニー(CANY)のフィルタを使用した場合の係数値を示す。ケニーのフィルタでは図のように係数

「0」があるので、この場合にはi=3番目の画素に対するフィルタリング処理(乗算処理)は不要である。そのため、このようなときは図15にその一部の構成を示すように乗算器MPY1に与えられる係数制御カウンタ160を工夫して、図16Bのように「0」のマスクデータが出力されないようにする。

【0062】その代わり図17のように画像データのアドレスAij(乗算時のアドレス)もコントロールされ、こうすることによってi=3番目の乗算処理を省くことができるため、乗算処理時間を短縮できる。

【0063】以上の(5×5)の演算処理は図4のように画像データを2分割し、それぞれを同時に処理することによって演算時間を1/2に短縮できる。そして、プロセッサエレメントPE自体はパイプライン処理であるためプロセッサエレメントPE自身における処理時間も短くて済む。

【0064】分割処理は図18のように画像データを3分割し、3分割された画像データMa, Mb, McをそれぞれのプロセッサユニットPUa~PUCで並列処理すれば全体の処理時間を短縮できる。

【0065】例えば各プロセッサユニットで3ライン分の画像データを一挙に処理するような場合を想定すると、この場合には図19のように各プロセッサユニットPUa~PUCに対して分割画像データMa~Mcそれぞれについてライン単位で3タイミング(1クロックに相当する)ごとに画像データのロードが行われると共に、3ライン分の画像データ(L1a~L3a), (L1b~L3b), (L1c~L3c)のロードが行われると同時に、画像データ(L1a~L3a)のロードが終了した次のタイミングから画像データ(L1a~L3a)の積和処理が実行される。他も同じである。

【0066】このような分割処理を行う場合には、1タイミングづつ順次遅れて3分割処理結果(演算処理結果)が得られるため、実質的に1クロックで画像処理できるようになる。

【0067】因みに、3ライン分の画像データのロードが終了してその積和処理を行い、積和処理が終了した段階で再び3ライン分の画像データをロードして同じような画像処理を行う場合には図19の3倍の処理時間がかかってしまう。

【0068】図19のような処理が行えるのも、この発明のように複数のラインメモリLMを持ち、任意のラインメモリLMに画像データがロードできるように構成され、そして演算結果が同一若しくは別のプロセッサエレメントPEにストアできるように構成され、データはパイプライン処理できるようになされているからに他ならない。

【0069】続いて、第3および第4の発明を説明する。図20は第3の発明の具体例であって、2つの画像メモリにストアされている画像データを論理積したり、

11

論理和する演算処理の場合に上述した汎用プロセッサを使用した場合である。

【0070】図20はそのうち論理積処理（アンド処理）の具体例であって、画像メモリ*i*、*j*にはライン情報としてランレングス化された画像情報がストアされているものとする。これら画像情報はそれぞれ同じプロセッサユニットPUに設けられたプロセッサエレメントPE1（*j*用）、PE2（*i*用）にロードされる。

【0071】そして、図21A～Cのように画像メモリ*i*の*i*ラインと画像メモリ*j*の*j*ラインのランレングスから両者のアンドをとる場合を例示する。両者のアンドをとると、同図Cのようなランレングスが得られる。

【0072】ランレングスコードに基づいて両者のアンドをとるには、図22のように各ランレングスの最初*s*と最後*e*のデータを基準にして両者の重なりを判断すればよい。両者の重なりと離間の状態を分類すれば図23に示すようにK1～K4の4種類となるので、両者のランレングス*s*、*e*を判断した結果、重なり具合がどの分類に属するかによって次の処理（アンド出力として利用するかどうか）が相違する。

【0073】ここで、ランレングスを利用したライン*i*、*j*の重なり具合を求めるには次のような判断式が用いられる。

【0074】 $a = js - ie - 1 \quad \dots (1)$

$b = is - je - 1 \quad \dots (2)$

$c = ie - je \quad \dots (3)$

この判断結果*a*、*b*、*c*から条件分岐K1～K4が決まる。

【0075】(1)～(3)式にあつて、*i*、*j*はそれぞれ別の画像メモリ*i*、*j*のデータであるから、上のような判断処理を行うには互いのプロセッサエレメントPEのデータを使用しなければならない。そのために同一のプロセッサユニットPU内のプロセッサエレメントPE同士でのデータの授受が必要になる。そこで、図2に示す入力切り替え手段MUX3を用いて図24のようなデータ経路を作つて上述した演算処理を実現させる。

【0076】まず、プロセッサエレメントPE1にあつては最初のタイミングで(1)式の算術を行う。そのため、累算器ALU1にはレジスタファイルRF1から*j*データ(=*js*)が、レジスタファイルRF2から*i*データ(=*ie*)がそれぞれリードされて(1)式が演算され、その結果はステータスレジスタSR1に保存される。

【0077】これと同じタイミングに、他方のプロセッサエレメントPE2にあつては(2)式の算術が行われる。そのため、累算器ALU2にはレジスタファイルRF1から*j*データ(=*js*)が、レジスタファイルRF2から*i*データ(=*ie*)がそれぞれリードされて(2)式が演算され、その結果はステータスレジスタSR2に保存される。

12

【0078】このような同時処理ができるのは、レジスタファイルRF1、RF2ともそのD、E端子から同時にデータ*s*、*e*をリードできるようになされているからである。

【0079】(1)、(2)式の演算が終了した次のタイミングでは、プロセッサエレメントPE1で(3)式の算術を行う。そのため、累算器ALU1にはレジスタファイルRF1から*j*データ(=*js*)が、レジスタファイルRF2から*i*データ(=*ie*)がそれぞれリードされて(3)式が演算され、その結果はステータスレジスタSR1に保存される。

【0080】ステータスレジスタSR1に保存された(1)式に関するステータス情報をSR0'とし、(2)式に関するものをSR1'、ステータスレジスタSR2に保存された(3)式に関するものをSR0'とすれば、これら3つのステータス情報SR0'、SR1'、SR0'がプロセッサエレメントPE3に設けられたシーケンサSEQ3の下位3ビットのアドレスA0～A2として使用される。

【0081】このアドレスはステータス情報に基づいて次に分岐すべき処理条件K1～K4を指定するためのマイクロプログラムのアドレスとして使用されるもので、指定された処理条件は同じプロセッサエレメントPE3のレジスタファイルRF3から出力される。

【0082】図25のように条件式(a)～(c)の結果と論理「0」、「1」とを関係付ければ、ステータス情報が例えば「011」（この論理値は(a)）

(c)、(b)の順でステータス情報を並べたときのものである)であつたときには、分岐条件K1が選ばれ、分岐条件K1とは図23に示すように*i*ラインと*j*ラインが互いに重なりあつていないときの処理条件で、このときはアンド出力もゼロとなる。

【0083】ステータス情報が「111」であつたときには分岐条件K2が選択され、この分岐条件K2のときは図23のように*j*ラインが*i*ラインの一部に重なっている状態を示すから、このときは図21Cのようなアンド出力が得られるように条件分岐K2のプログラムが組まれている。

【0084】このように異なるプロセッサエレメントPE1、PE2で得られたステータス情報を組み合わせてマイクロプログラムに対する1つのアドレスとして直接使用できるように構成すれば、ステータス情報を一旦シフトレジスタなどに保存し、全てのステータス情報が揃った段階で始めてマイクロプログラムを起動させるようにする場合よりも処理時間を短縮でき、パイプライン処理の効果が一層大きくなる。

【0085】この発明に係るプロセッサは汎用性があるので、上述したようにプロセッサユニットPUを増やすだけで積和演算機能の拡張が簡単に行える他、同一のプロセッサユニットPU内でもプロセッサエレメントPE

同士のデータを使用した演算処理ができる。これらの他に、このプロセッサは画像メモリに対する制御回路としても使用することができる。

【0086】第4の発明は汎用プロセッサを画像メモリ用制御回路として使用した場合であって、図26はそのときの一例を示す系統図である。

【0087】使用する画像メモリとしてはSRAMと一対のDRAMを例示し、汎用のプロセッサユニットPUは1個使用され、プロセッサエレメントPEにそれぞれ設けられたレジスタファイルRF1、RF2がアドレス発生器として機能する。

【0088】SRAMはレジスタファイルRF1でその下位アドレスLADRSが、RF2で上位アドレスHADRSが指定される。CONT端子にはレジスタファイルRF1からライトイネーブル信号が与えられる。SRAMの画像データはレジスタR4を制御してラインメモリLMにロードされるか、若しくはレジスタファイルRF1、RF2、RF3に直接ロードされる。

【0089】DRAMa、DRAMbは高速でアクセスするためにリード、ライトが交互に行われる。DRAMaはレジスタファイルRF1で制御され、DRAMbはレジスタファイルRF2で制御される。それぞれのCONT端子には縦横方向の各アドレスを指定するRAS、CASおよびライトイネーブル信号が加えられる。

【0090】ラインメモリLMよりロードされた画像データはバッファレジスタBF1、BF2を介して対応するDRAMに与えられてこれがライトされる。DRAMa、DRAMbよりリードされた画像データはバッファレジスタBF3、BF4を介してラインメモリLMにロードされる。

【0091】レジスタR3とR4はSRAMとDRAMのデータの何れかを選択するためのもので、デコーダDECによってコントロールされる。デコーダDECにはそのコントロール信号がレジスタファイルRF1から供給される。プロセッサエレメントPE3のレジスタファイルRF3はラインメモリLMを選択するためのものである。

【0092】図27はDRAMを制御するときのタイミングチャートの一例で、同図A~CのようにプロセッサエレメントPE1からDRAMaに対して偶数番目のアドレスが指定され、また同図F~HのようにプロセッサエレメントPE2からは1クロックシフトさせてDRAMbに対して奇数番目のアドレスが指定される。

【0093】そして、同図Jのように基準クロックの2倍周期のリードクロックをそれぞれのメモリに与えると同図E、Iのタイミングで画像データがリードされるから、最終的には同図Kに示すように基準クロックと同じタイミングに画像データが順次出力される。

【0094】上述した実施例はこの発明の一例であって、列挙したこれらの実施例以外でもこの発明を利用

することは言うまでもない。

【0095】

【発明の効果】以上のように、第1~第4の発明に係る汎用プロセッサでは、積和演算機能を有する複数のプロセッサエレメントで構成され、これらの複数のプロセッサエレメント同士は互いに別のプロセッサエレメントの任意の演算処理ステップにおけるデータを、自己のプロセッサエレメントにおける任意の処理ステップに取り込むことができるようになされたことを特徴とするものである。

【0096】これによれば、この汎用プロセッサの多用途化を実現できる。つまり、パイプライン処理を行っているため、積和演算処理用のプロセッサとして使用できるし、汎用プロセッサを複数個使用することによって積和演算処理の拡張が容易に実現できる他、同じプロセッサ内でも複数のプロセッサエレメント同士のデータの授受が可能であるため、2以上の画像メモリに対する画像処理なども簡単に実現できる特徴を有する。

【0097】この発明では実施例以外の画像処理でも、使用する個数、組合せなどを工夫することによってその目的に合致したプロセッサを実現できる。

【図面の簡単な説明】

【図1】この発明に係る汎用プロセッサの概略構成を示すブロック図である。

【図2】汎用プロセッサの具体例を示す系統図である。

【図3】複数の汎用プロセッサを使用したときの基本的な組合せを示すブロック図である。

【図4】積和演算処理用に使用したときの汎用プロセッサのブロック図である。

【図5】画像データの配列例を示す図である。

【図6】フィルタ係数の配列を示す図である。

【図7】積和演算処理の説明に供するブロック図である。

【図8】積和演算処理の説明に供するブロック図である。

【図9】積和演算処理の説明に供するブロック図である。

【図10】1つのプロセッサユニットにおける積和演算処理の説明図である。

【図11】(5×5)の加算処理例を示すブロック図である。

【図12】加算処理動作を説明する図である。

【図13】プロセッサユニットPU4での処理動作の説明図である。

【図14】マスクデータの一例の図である。

【図15】特定のマスクデータを使用したときの乗算処理例を示すブロック図である。

【図16】そのときの説明図である。

【図17】乗算動作とそのときのアドレスの関係を示す図である。

15

【図 18】画像データの分割処理例を示すブロック図である。

【図 19】分割処理用のタイミング図である。

【図 20】画像メモリの説明図である。

【図 21】アンド処理の説明図である。

【図 22】ランレングスコードの説明図である。

【図 23】条件分岐の説明図である。

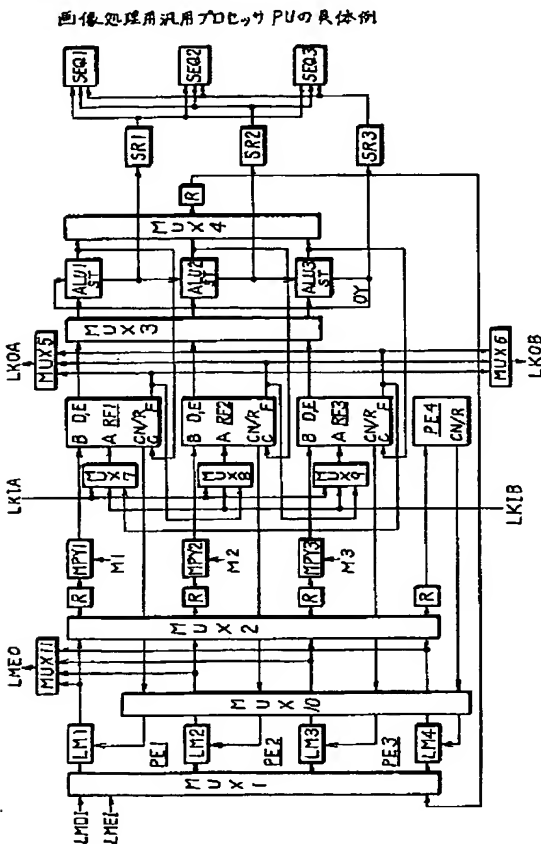
【図 24】プロセッサエレメント間の演算処理例を示すブロック図である。

【図 25】その動作説明に供する図である。

【図 26】画像メモリ用制御回路の一例を示すブロック図である。

【図 27】その動作説明に供する波形図である。

【図 2】



16

*【符号の説明】

PU (PU1~PU6) 汎用プロセッサ (プロセッサユニット)

PE (PE1~PE3) プロセッサエレメント

LM (LM1~LM4) ラインメモリ

MPY (MPY1~MPY3) 乗算器

RF (RF1~RF3) レジスタファイル

ALU (ALU1~ALU3) 累算器

SR (SR1~SR3) ステータスレジスタ

10 SEQ (SEQ1~SEQ3) シーケンサ

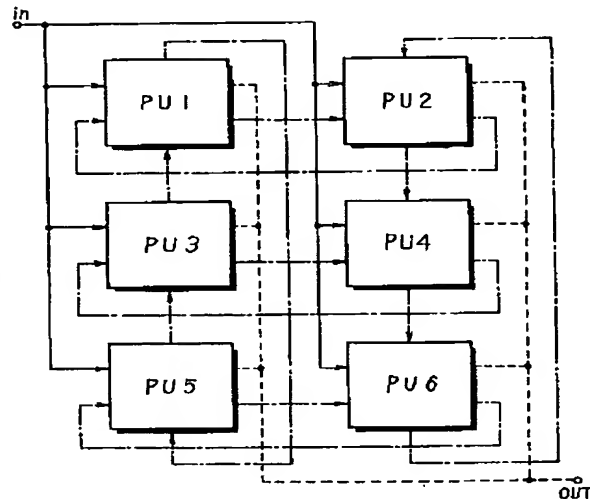
MUX (MUX1~MUX11) 入力切り替え手段

R (R1~R4) レジスタ

* BF (BF1~BF4) バッファレジスタ

【図 3】

汎用プロセッサ PU



【図 5】

画像データの配列

	→ 画素				
a	D00	D01	D02	D03	D04
b	D10	D11	D12	D13	D14
c	D20	D21	D22	D23	D24
d	D30	D31	D32	D33	D34
e	D40	D41	D42	D43	D44

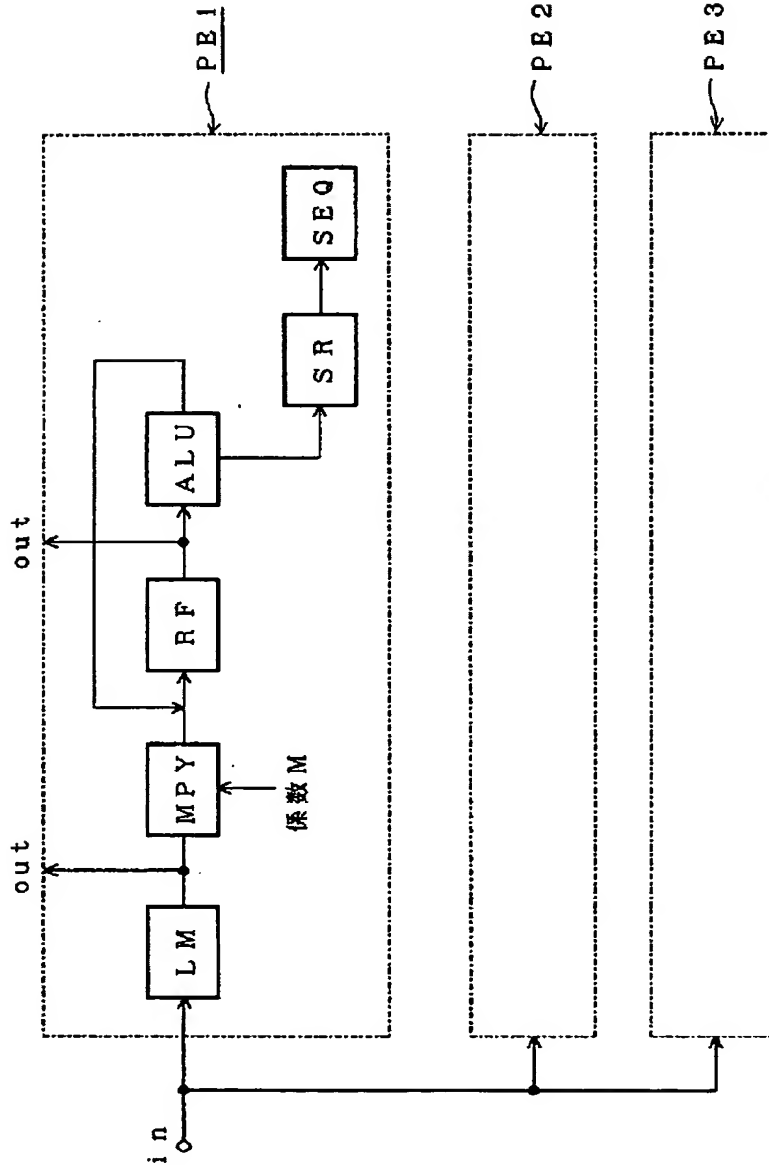
【図 6】

(5×5) サイズのフィルタ係数

M00	M01	M02	M03	M04
M10	M11	M12	M13	M14
M20	M21	M22	M23	M24
M30	M31	M32	M33	M34
M40	M41	M42	M43	M44

【図1】

画像処理用汎用プロセッサPU



【図10】

ラインにおけるPU2での演算処理

処理 ステップ	時点	t0 t1 t2 t3 t4	t5 t6 t7 t8 t9	t10 t11 t12 t13 t14	t15 t16 t17
a	MPY1 の処理	D00xM00→R0 D01xM01→R1 D02xM02→R2 D03xM03→R3 D04xM04→R4	D01xM00→R5 D02xM01→R6 D03xM02→R7 D04xM03→R8 D05xM04→R9	D02xM00→R0 D03xM01→R1 D04xM02→R2 D05xM03→R3 D06xM04→R4	
b	ALU1 の処理 RF1の C端子から のロード		R0+R1→R1 R2+R3→R2 R1+R3→R3 R3+R4→R10a	R5+R6→R5 R7+R8→R6 R8+R9→R8 R8+R9→R11a	R0+R1→R1 R2+R3→R2 R1+R3→R3 R3+R4→R12a
c	RU4の RF1へ ロード			R10a出力	R11a出力

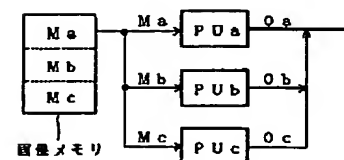
【図14】

マスキングデータの例

-1	-3	0	3	1
-7	-15	0	15	7
-11	-24	0	24	11
-7	-15	0	15	7
-1	-3	0	3	1

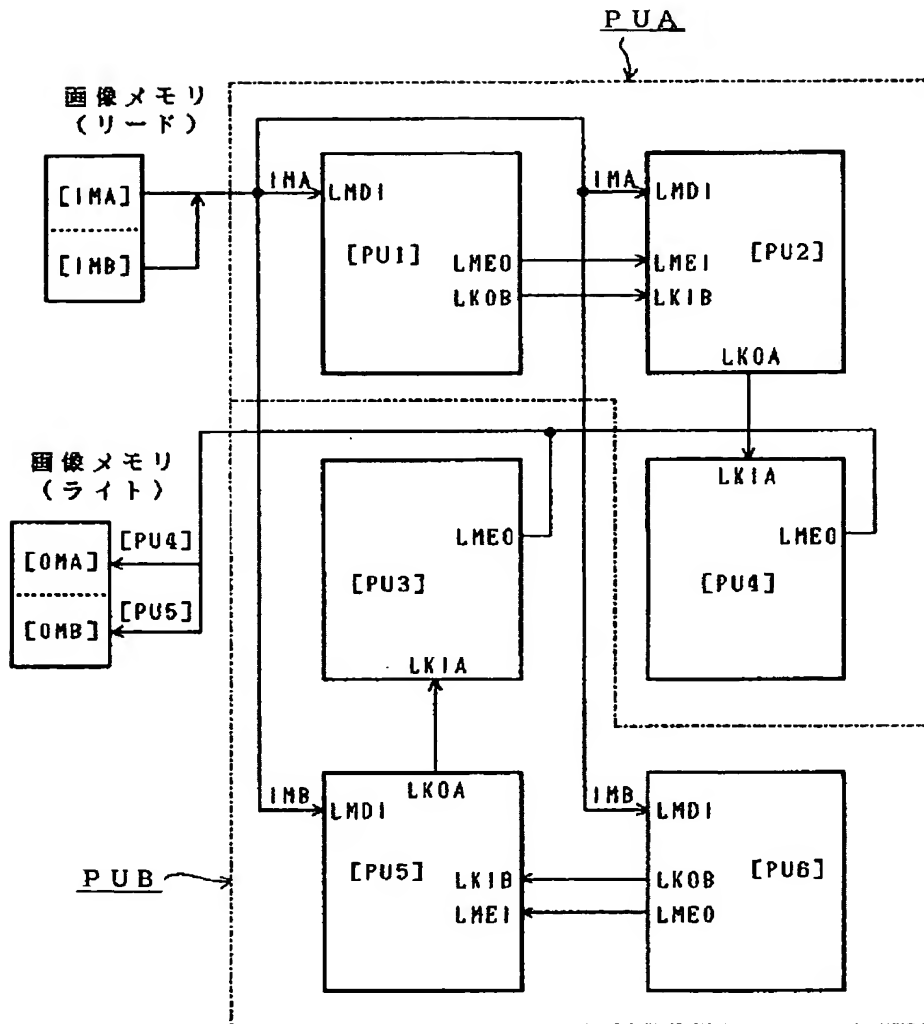
【図18】

画像データの分割処理例



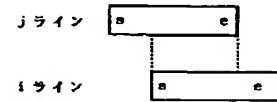
【図4】

(5×5)の積和演算用プロセッサ

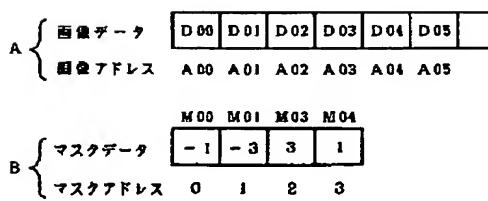


【図22】

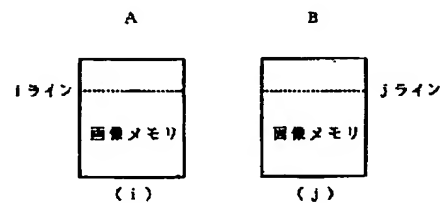
ランレングスコードの例



【図16】

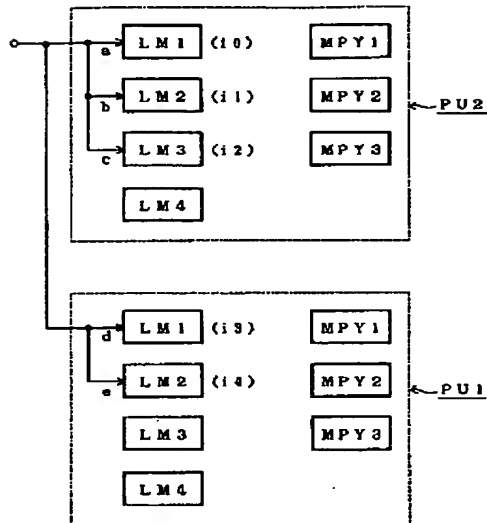


【図20】



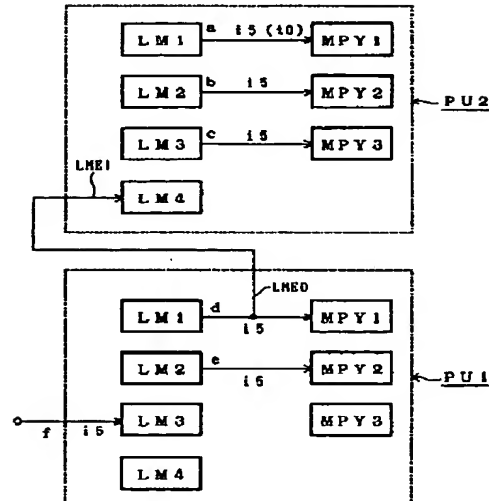
【図7】

(5×5)の積和演算処理例(その1)



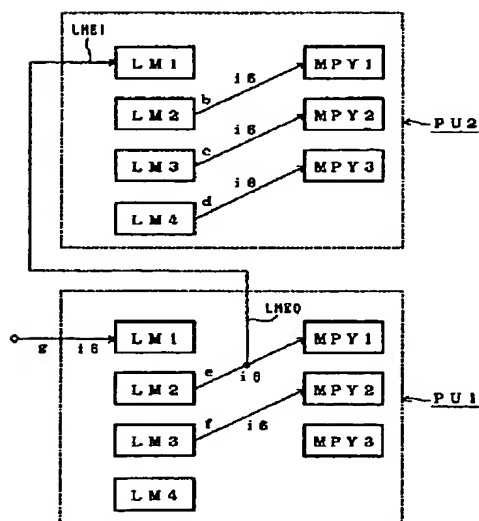
【図8】

(5×5)の積和演算処理例(その2)



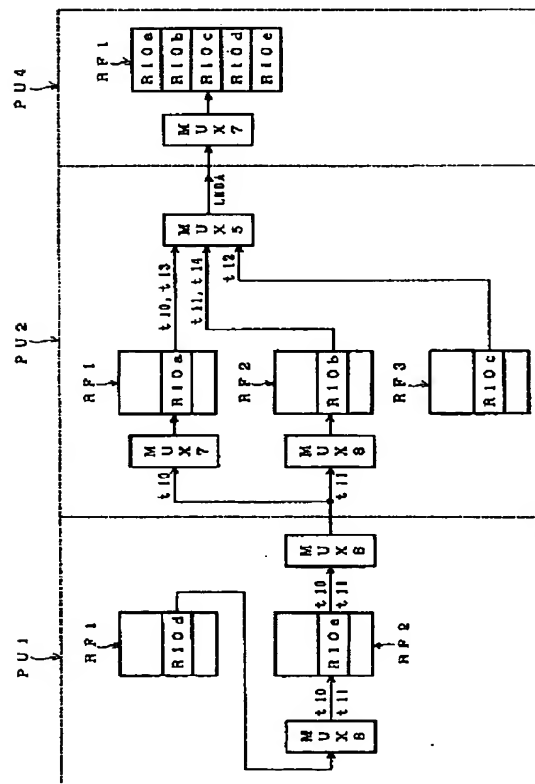
【図9】

(5×5)の積和演算処理例(その3)



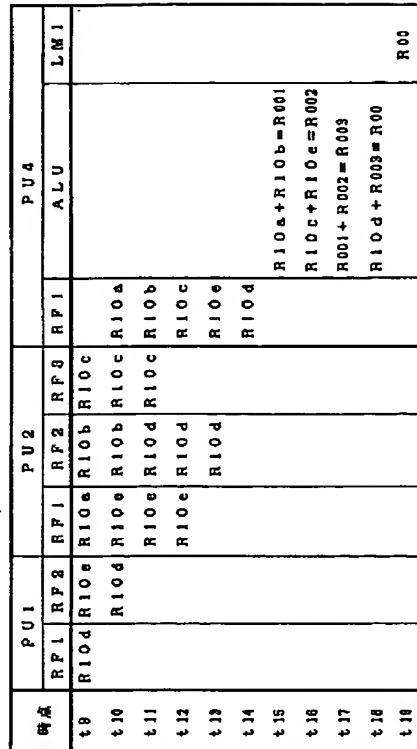
【図11】

(5×5)の加算処理説明図



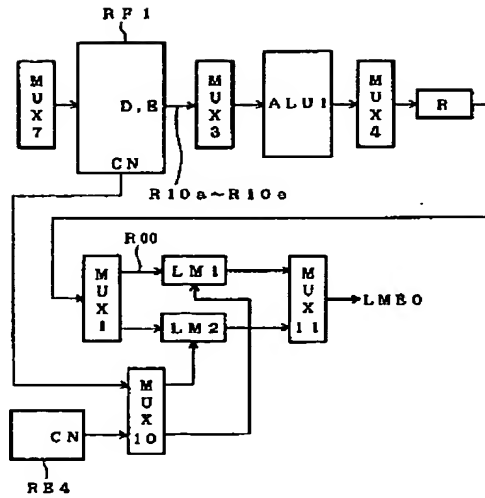
【図12】

加算動作の説明



【図13】

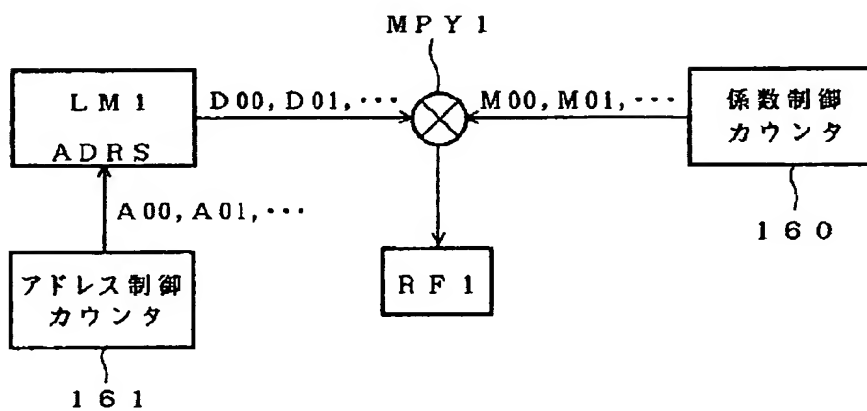
PU4での処理動作説明図



【図17】

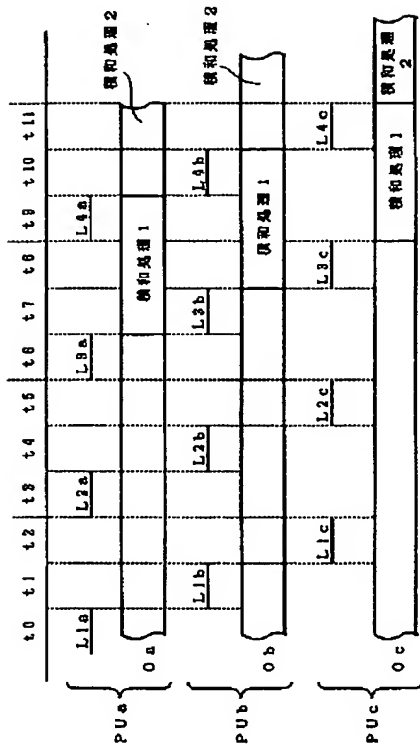
アドレス値	アドレス	MPY1
0	A00	D00×M00
+1	A01	D01×M01
+2	A03	D03×M03
+1	A04	D04×M04
-3	A01	D01×M00
+1	A02	D02×M01
+2	A04	D04×M03
+1	A05	D05×M04

【図15】



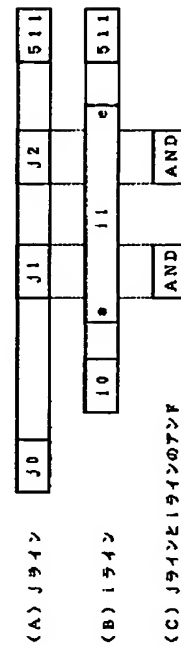
【図19】

分岐処理段可用タイミングチャート



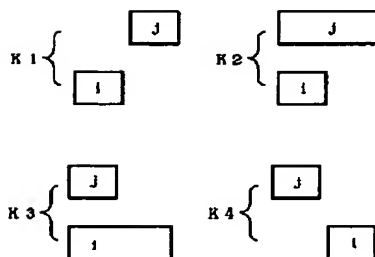
【図21】

1ライン分のランレングスコードからのアンド図例



【図23】

条件分岐の説明



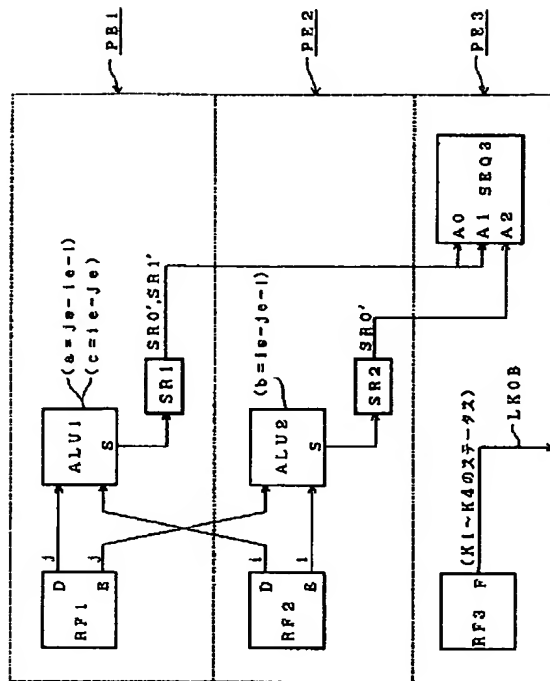
【図25】

ステータス情報を用いた条件分岐例

プロセッサ		P E 1		P E 2
ステータス		(a) のステータス	(c) のステータス	(b) のステータス
ステータスレジスタ		S R 0'	S R 1'	S R 0'
シーケンサSEQ3へのアドレス		A 0	A 1	A 2
分岐条件	K 1	0 (a > 0)	1 (c < 0)	1 (b < 0)
	K 2	1 (a ≤ 0)	1 (c ≤ 0)	1 (b < 0)
	K 3	1 (a < 0)	0 (c > 0)	1 (b ≤ 0)
	K 4	1 (a < 0)	0 (c > 0)	0 (b > 0)

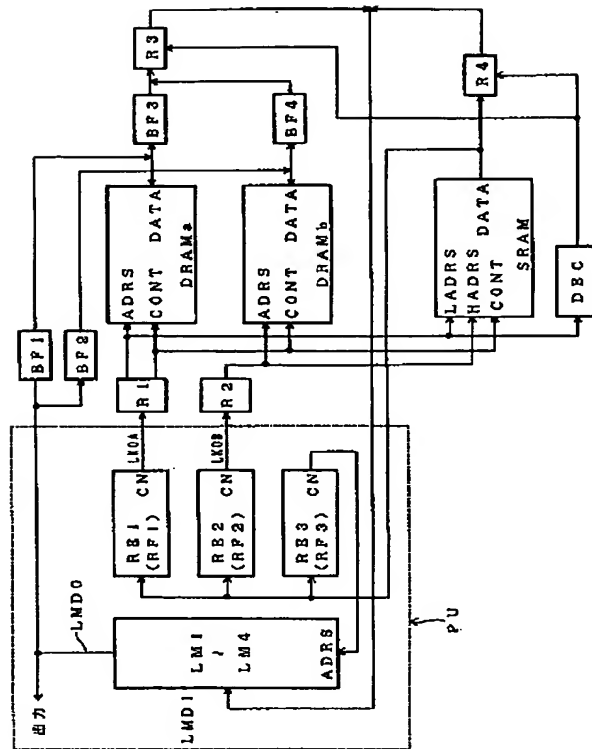
【図24】

PE間伝送経路例



【図26】

断線メモリ用制御回路



【図 27】

DRAMを読出すときのタイミングチャート

